# METHOD AND APPARATUS FOR DATA LOGGING

This invention relates to a method and apparatus for data logging.

## BACKGROUND

Data logging in this specification is the process of collecting data from mobile devices performed in order to obtain business information relating to how the mobile devices operate. For example, position, location and speed of a vehicle over time is useful log data for use in insurance liability calculations for that vehicle. In another example, signal strength of a mobile communication system over both time and position is useful log data to enable a telecommunication company to plan its transmitter locations. Such data is collected by a mobile embedded system using positional information and signal strength information from sources including the network itself (e.g. GSM) and global positioning satellites (GPS). Log data is stored in the mobile embedded system for later transmission to the central system. Transmission is by mobile phone network or other wireless technology.

Transmission of the data log may be performed on demand; when the device is ready it requests control of the transmission channel. Such a system is described in US Patent Publication 6263268 which discloses a mobile automotive telemetry system for installation on-board a vehicle. It includes a diagnostic structure for monitoring operational functions of a vehicle and a server which communicates with the diagnostic structure to receive operational information. The operational information is uploaded to the server when the information is ready.

Another download on demand system, International Patent Publication 02/03350, discloses a method and system for monitoring cellular communication. The method continuously extracts traffic load and speed on roads within the coverage area of a cellular network from a mobile device in a vehicle. The data is extracted directly from the higher level of communication in a cellular network so there is no scheduled or negotiated download of data from the mobile device.

One problem with downloading log data on demand is that it can lead to a conflict situation when several devices are requesting control of a single channel and attempting to download at the same time. Only one request per channel will be successful at any one time and the other requests fail. The failed requests use download resource so that more resource is used for non-ordered requests than for ordered requests. One way to order the downloads is to schedule them to come in at a certain times.

US Patent publication 0028313 discloses a distributed telemetry method and system affected by co-ordinating the taking of readings of a parameter by mobile phone users, the parameter readings being sent to a service system together with location information on the users. It is the task of a query scheduler to, amongst other things, organise when the reading of interest are to be taken. The reading is sent to the service system immediately or triggered by, for example, a scheduled time.

The problem with scheduled remote data logging is that simultaneous and multiple device upload of data can create overload on a server that collects such log data when the download size is different from that scheduled.

**SUMMARY OF INVENTIONS**

According to a first aspect of the present invention there is provided a data logging method for transferring data from a plurality of client devices to a server, said method comprising:

building a schedule of transfer periods based on an estimated transfer size for each device;

receiving an actual transfer size for a device;

updating the schedule for all devices with respect to the difference in the received actual transfer size and the corresponding estimated transfer size for said device; and

transferring data for said device.

Advantageously said step of building a schedule of transfer periods comprises:

estimating a future transfer size for a device;

calculating a transfer period when the device is scheduled to download its data to the server based on that device's future transfer size estimate and other devices' transfer periods;

storing the transfer periods and a corresponding device reference in a data structure; and

performing the above steps with respect to each device.

More advantageously the step of updating the schedule comprises: re-calculating the transfer period for the device based on the actual transfer size.

Prefereably the step of updating the schedule further comprises: re-calculating transfer periods of other devices in the schedule if the re-calculated transfer period of said device effects the transfer periods of the other devices.

More preferably, if the originally calculated transfer period differs from the re-calculated transfer period, one or more subsequent transfers may be re-scheduled.

Suitably the future transfer size is an estimate based on a client's historic transfer size.

More suitably the future transfer size is acquired from the client based on the present size of the log data.

Advantageously the future transfer size is an estimate based on the client's historic transfer size and the present size of the log data.

## DESCRIPTION OF DRAWINGS

In order to promote a fuller understanding of this and other aspects of the present invention, an embodiment of the invention will now be described, by means of example only, with reference to the accompanying drawings in which:

Figure 1 is a schematic diagram of the present embodiment of the invention;

Figure 2 is a schematic diagram of a profile data structure stored by the present embodiment of the invention;

Figure 3 is a schematic diagram of a plan data structure stored by the present embodiment of the invention;

and

Figure 4 is a method according to a second embodiment of the present invention.

## DESCRIPTION OF THE EMBODIMENTS

Referring to Figure 1 a data logging system comprises: a server 300 connected over a mobile network to a plurality of remote client devices 100A...N. The first device is 100A, second device is 100B and so on up to 100N where N is of order a million devices. Each device 100A...N comprises: profile data 102; a device profiler 104; a data log 106; a device controller 108; and a data exchange 110. The server 300 comprises: profile data 301; a device profiler 302; a scheduler 304; a plan 306; an updater component 308; data exchange 310; a bandwidth forecast component 311; and an upload component 312. Log data is stored in a datastore 400.

Device profiler 302 maintains each device profile 102 collected from the client devices.

A device profile 102 includes characteristics relating to the device but not the download data itself. Referring to Figure 2, device profile 102 comprises two profiles for each device in the preferred embodiment: firstly a connectivity profile 103A; and secondly a download profile 103B.

The connectively profile 103A includes GSM radio reception power over time and comprises a data structure having a date and time field and GSM signal field. In a further embodiment the geographical position of the device will be included in the connectivity profile where it is derived from global positioning system (GPS) data or trigonmetric data from the GSM receivers. The status of a device is recorded over a week as this is mostly likely to show a pattern. However, in other embodiments, a longer period may be used instead of or as well as. A day of data is normally regarded as the minimum, although theoretically it could be smaller, and three weeks of data gives better averages. More than four weeks of data puts pressure on the storage resource of the device profiler 302.

The quantity of data previously collected allows for at least an estimate to be made of the next quantity of data downloaded. The device profiler 302 provides information to the device scheduler 304 to enable it to establish an estimate for connection time. It also provides information relating to GSM power levels so that unsuitable connection times can be estimated.

The download profile 103B, stores, for each device, a record of each download of data and comprises the time of download and the quantity of data collected in the download.

The scheduler 304 builds the plan 306 by allocating time periods to each specific device based initially on the amount of data that each device is expected to transfer. Device scheduler 304 receives the actual network usage from the data exchange 310 and bandwidth forecast information from component 311. If the scheduler 304 sees that current network usage exceeds or is much less than that used to build the plan 306 then the scheduler 304 updates the plan 306. The device scheduler 304 works to substantially 80% full capacity so that overruns can be catered for and rescheduling work does not continuously replan.

The plan 306 (see Figure 3) is a data structure that stores a download schedule for each device. Each download for a device is a record in the database having four fields: 1) transfer period (start time and end time); 2) the device identification; and 3) the transfer size.

The updater component 308 keeps a device updated as to its scheduled time by passing messages with the current schedule details and also when relevant changes are made to the plan 306. It keeps track of a device that is off-line and informs the off-line device as soon as it becomes on-line through the data exchange 310.

The update component 308 negotiates with the scheduler 304 in case the device has run out of memory or has not downloaded for an excessive period of time. The scheduler 304 identifies change in the plan 306 and informs the updater component 308 and updates the plan 306 with confirmation from the devices 100A...N.

The bandwidth forecast component 311 monitors current download activity from actual data transfers going through the data exchange 310. From this information the present download bandwidth can be monitored and stored for future planning reference. This data is then used to provide a forecast of network capacity for the scheduler 304, which then in turn may choose to throttle back the data transfer by moving devices, or move devices up to take advantage of available bandwidth. In another embodiment the bandwidth component acquires forecast information from a network supplier.

The uploader 312 determines when to update client devices with new software. It takes as input plan 306 to provide an indication of other traffic that may make use of the data communications lines 210,206. The plan 306 will have an impact on deciding when to upload software to the client devices as each download will reduce the 'effective' communications bandwidth for data upload.

The client device 100A will now be described. Client devices 100B..N have the same components and configuration but different identification. Each device 100A...N comprises: profile data 102; a device profiler 104; a data log 106; a device controller 108; and a data exchange 110.

The profile data 102, maintained by the device profile component 104, maintains a profile of the device's connectivity and data volumetrics. It is the profile data 102 that is sent to the server 300 to be used in planning and prioritization by scheduler 304.

The data log 106 contains the log data for transfer to the server 300, it also may contain any specific data used by the device controller 108. The key objective is to transfer log data 106 from the client to the datastore 400 via the server 300.

Device controller 108 is responsible for ensuring co-ordination of the log data 106 and controls the download of data to the server 300 and the communication of data volumes. It is the device controller 108 that initiates the data connection and bulk transfer based on the scheduled time received from the server 300. The device controller 108 receives the schedule information from plan 306 via the update component 308. Before the downloading of the log data the device controller 108 establishes communication with the scheduler 304 through update component 308 and the data exchange mechanism 310/110 to check for final adjustments. Ideally each client 100A...N would be controlled by the same version of device controller 108 but it may be that some devices have been updated by uploader 312 and others are using an older version. Software updates can be transferred between the server 300 and the clients 100A...N along the same communication lines as the data is transferred.

In the preferred embodiment, the method re-schedules devices in the plan if estimated download sizes differ from the actual download size.

Method 500 of this further embodiment is described below with reference to Figure 4.

The scheduler 304 selects, step 502, a device from the profile data 301. The selection is initially in order of prior transfer size.

The scheduler 304 estimates, step 504, a future transfer size for each device by looking up the download profile of the device and using the previous transfer size. An average of previous transfer sizes may be used.

The scheduler 304 calculates, step 506, a transfer period based on the estimated transfer size.

The estimated transfer period is stored, step 508, in plan 306 and the updater 308 sends the scheduled time to the device via the data exchange mechanism 310.

Step 510 returns the process to step 502 so that steps 502 to 508 are repeated for all devices in the device profile.

Once the plan is completed, each device will have received a scheduled transfer time from the updater 308. Each device will begin the download of log data at the scheduled time. However, the server has to recalculate the scheduled time for some devices if the estimated transfer size is not the same as the actual size.

Prior to transfer, a scheduled device contacts, step 512, the server at the scheduled time with the actual download size.

Scheduler 304 acquires, step 514, the actual transfer size.
If the actual transfer size is different from the estimated one then the different transfer period will change the scheduled times of other devices. At step 516, the scheduler 304 re-calculates the transfer periods of a device or devices affected.

The scheduler 304 re-schedules, step 518, the present device if the actual transfer period of the present device is different from what was estimated in the plan. If the transfer period is too long then the next scheduled device will either be pushed forward in time or substituted for a different device having a shorter transfer period. In this embodiment a device with a smaller transfer period is substituted for the next device which advantageously minimizes change to the whole plan 306. In another embodiment the next scheduled device is given a new transfer period which will affect subsequent downloads but retain the original order. If the transfer period is too short there becomes available some free resource, the free resource is filed with a new device having a transfer period which will fill it. In this case it would also be possible to bring forward all the device transfer periods but keeping to substantially the original plan is preferred. In another embodiment the present device itself may be rescheduled in a way which minimizes change to the plan 306.

After or during the previous step, the present client transmits, step 520, its logged data.

The process is repeated, step 522, for all devices in the plan 306.

In the preferred embodiment all the profile data is stored in the server 300 but in another embodiment the data could be stored on the device itself. The profile information is then requested from the device when it is needed. In this other embodiment the data may not be available exactly when it is required so it is not preferred.

The server 300 initially creates a plan 306 for all clients based on a simple staggered algorithm.

Before the initial plan is created, all the device controllers 108 notify the device profiler 302 of the quantity of data to be sent. The updater 308 informs the device controller 108 of the time to connect to the server 300 to transfer data 106.

The profiler 302 will store this profile and pass information on to the dynamic rescheduler 304 which will use the existing plan 306 and if needed adjust the plan and re-notify the client via 308.

The client also sends its profile data 102 via the device profile module 104 to the server device profile 302. The profile data 102 stores the quantity of data gathered per time unit and time of available network coverage.

The scheduler 304 uses historic profile data from the profile data 304 to plan arrival times and connection length to spread the load out during the working day. The optimal plan being to keep a core number of clients communicating but not to overload the system. To do this known data volumes sent by the client scheduler 108 and the predicted volumes from the device profiler 302 are used. This information is used along with the actual and predicted network band with (from 310 bandwidth forecaster).

Although the embodiments have been described in terms of a single server it is possible to scale the solution up to two or more servers.